In the claims:

Please amend claims 1, 2, 6-11, 13, 25, 27 and 33. The status of the claims is as follows:

1. (Currently Amended) A method for optimizing a solution set for a problem comprising the sequential steps of:

a) generating a first set of solutions comprising a plurality of members;

b) creating a second set of solutions using a selection operator to select members from said first set of solutions;

c) ~~creating a probabilistic model~~ fitting said second set of solutions with a probabilistic model, said probabilistic model allowing for multiple variables to be merged together to form a single new variable whereby hierarchically decomposable problems can be modeled;

d) using said model to generate a third set of solutions; and,

e) replacing at least a portion of said first set of solutions with said third set of solutions to create a new solution set.

2. (Currently Amended) A method for optimizing a solution set as in claim 1 wherein the step of ~~creating a model~~ fitting said second set of solutions with said probabilistic model comprises creating a plurality of models that fit said second set of solutions, selecting one model from said plurality of models, and using said one model to create said third set of solutions.

3. (Original) A method for optimizing a solution set as in claim 2 wherein the step of selecting one model comprises using a scoring metric to score each of said models and searching said scored models to determine the highest scoring of said plurality of models.

4. (Original) A method for optimizing a solution set as in claim 3 wherein said scoring metric is chosen from the group of scoring metrics consisting of a minimum description length metric, and the Bayesian-Dirichlet metric; and wherein the step of searching all of said scored networks comprises using a search method chosen from the group consisting of a greedy algorithm method, a local hill climbing method, a gradient search, a tabu search, and a simulated annealing method.

5. (Original) A method for optimizing a solution set as in claim 2 wherein the step of selecting one model comprises comparing said plurality of models through a binary relation to define a better than relation between said models.

6. (Currently Amended) A method for optimizing a solution set as in claim 1 wherein the step of fitting said second solution set with ~~creating~~ a probabilistic model comprises creating a model that models dependencies, independencies, conditional dependencies, and conditional independencies on a plurality of hierarchical levels between at least a portion of said merged variables.

7. (Currently Amended) A method for optimizing a solution set as in claim 1 wherein the step of fitting said second solution set with ~~creating~~ a probabilistic model ~~fitting said second set of solutions~~ comprises creating one or more models chosen from the group consisting of Bayesian networks with local structures, and Huffman networks.

8. (Currently Amended) A method for optimizing a solution set as in claim 1 wherein the step of fitting said second solution set with ~~creating~~ a probabilistic model comprises creating a probabilistic model that utilizes local structures to represent conditional probabilities between variables.

9. (Currently Amended) A method for optimizing a solution set as in claim 1 wherein the step of fitting said second solution set with ~~creating~~ a probabilistic model ~~fitting said second set of solutions~~ comprises creating a Bayesian network model comprising one or more features to model a hierarchal problem decomposition, said one or more features selected from the group consisting of local structures, hidden variables, and histograms.

10. (Currently Amended) A method for optimizing a solution set as in claim 1 wherein the step of ~~creating a model~~ fitting said second set of solutions with a probabilistic model comprises creating a Bayesian network model comprising local structures selected from the group consisting of default tables, decision trees, and decision graphs.

11. (Currently Amended) A method for optimizing a solution set as in claim 1 wherein the step of fitting said second solution set with ~~creating~~ a probabilistic model comprises building a network to model a probability distribution of said second set of solutions, said network representing a plurality of merged variables as a single variable, said network representing dependencies and independencies between at least a portion of said merged variables over a plurality of hierarchical levels.

12. (Original) A method as in claim 11 wherein the step of building said network comprises building said network using a scoring metric and a search method to optimize said network.

13. (Currently Amended) A method for optimizing a solution set as in claim 1 wherein the step of fitting said second solution set with ~~generating~~ a probabilistic model ~~fitting said second set of solutions~~ comprises generating a plurality of probabilistic models fitting said second set of solutions, each of said models allowing for multiple variables to be merged together to form a single new variable whereby hierarchically

decomposable problems can be modeled; and wherein the step of using said probabilistic model to generate a third set of solutions comprises using each of said plurality of models to generate a portion of said third set of solutions.

14. (Original) A method for optimizing a solution set as in claim 13, wherein the method further comprises the step of using a first of said plurality of models to generate a selected percentage of said third set of solutions, and using a second of said plurality of solutions to generate a selected percentage of said third set of solutions.

15. (Original) A method for optimizing a solution set as in claim 1 wherein the method further comprises a step of performing niching to preserve diversity of solutions.

16. (Original) A method for optimizing a solution set as in claim 1 wherein said selection operator provides for niching to preserve diversity among said members of said second set.

17. (Original) A method for optimizing a solution set as in claim 1 wherein said selection operator comprises a restricted tournament selection operator.

18. (Original) A method for optimizing a solution set as in claim 1 wherein said selection operator comprises one or more operators selected from the group consisting of a tournament selection operator, a truncation selection operator, a fitness selection operator, a preselection operator, a crowding selection operator, a clustering selection operator, a lambda-mu selection operator, and a Boltzmann selection operator.

19. (Original) A method for optimizing a solution set as in claim 1 wherein the step of replacing at least a portion of said first solution set with said third

solution set further comprises replacing at least a portion of said first solution set with said third solution set using a replacement operator that provides for niching.

20. (Original) A method for optimizing a solution set as in claim 19 wherein said replacement operator performs niching using a niching method selected from the group of methods consisting of tournament replacement and crowding.

21. (Original) A method for optimizing a solution set as in claim 20 wherein said replacement operator comprises restricted tournament replacement.

22. (Original) A method for optimizing a solution set as in claim 1 wherein the step of replacing at least a portion of said first set of solutions comprises performing niching to preserve diversity of said solutions through the steps of:

subdividing said first set of solutions into a plurality of sub-sets;

comparing one of said third set of solutions with members of one of said sub-sets; and

replacing one of said members of said sub-sets with said one of said third set of solutions if said one of said third solution set is superior to said members of said one of said sub-sets.

23. (Original) A method for optimizing a solution set as in claim 22 wherein each of said sub-sets have between 15-25 members.

24. (Original) A method for optimizing a solution set as in claim 22 wherein each of said sub-sets have a number of members greater than 25 and proportional to the size of the problem being solved.

25. (Currently Amended) A method for optimizing a solution set as in claim 1 wherein the step of creating a second set of solutions from said first set of

solutions comprises creating a plurality of second sets of solutions, wherein the step of fitting said second set of solutions with ~~creating~~ a probabilistic model ~~fitting said second set~~ comprises fitting each of said plurality of second sets of solutions with ~~creating~~ a probabilistic ~~model fitting each of said second sets~~, wherein the step of using said model to generate a third set of solutions comprises using each of said probabilistic models to generate a plurality of third sets of solutions, and wherein the step of replacing at least a portion of said first set of solutions with said third set comprises replacing at least a portion of said first set with each of said third sets of solutions.

26. (Original) A method for optimizing a solution set as in claim 1 wherein the method further comprises a step of:

evaluating said new solution set to determine if completion criteria are satisfied, and if said completion criteria are not satisfied replacing said first set of solutions with said new set of solutions and repeating the steps of b) through e) until said completion criteria are satisfied.

27. (Currently Amended) A method for optimizing a solution set as in claim 1 wherein the step of using a selection operator to create a second set of ~~members~~ solutions from said first set further comprises using a fitness function to determine a fitness for each of said members, and wherein said selection operator selects members having a high fitness.

28. (Original) A method for optimizing a solution set as in claim 27 wherein said fitness function comprises one or more fitness functions selected from the group consisting of a co-evolved fitness function, a multi-objective fitness function, a multi-modal fitness function, an interactive fitness function, an approximate fitness function, a simulation fitness function, and an execution fitness function.

29. (Original) A method for optimizing a solution set as in claim 1 wherein the step of generating a first set of solutions comprises generating said first set of solutions using a probability distribution biased by prior knowledge.

30. (Original) A method for optimizing a solution set as in claim 1 wherein said members of said first solution set are coded in a format selected from the group consisting of fixed and variable length strings.

31. (Original) A method for optimizing a solution set as in claim 30 wherein said members of said first solution set are in a format that comprises one or more formats selected from the group consisting of: binary code, k-ary code, floating point code, alphanumeric code, real vectors, logical expressions, program codes, images, random keys, and rules.

32. (Original) A method for optimizing a solution set as in claim 1 wherein the method further comprises the preliminary step of converting data into said first solution set.

33. (Currently amended) A method for optimizing a solution set for a problem comprising the steps of:

generating a first set of solutions comprising a plurality of members;

creating a second set of solutions by selecting a sub-set of members from said first set of solutions;

~~creating a probabilistic model~~ fitting said second set of solutions with a probabilistic model;

using said model to generate a third set of solutions; and,

replacing at least a portion of said first set of solutions with said third set of solutions to create a new solution set, said replacing comprising niching to preserve diversity among members of said new solution set.

34. (Original) A method for optimizing a solution set as in claim 33 wherein the step of replacing at least a portion of said first set of solutions with said third set with niching is performed using one or more operators chosen from the group consisting of a tournament operator, a crowding operator, a pre-selection operator, a clustering operator, and a fitness sharing operator.

35. (Original) A method for optimizing a solution set as in claim 33 wherein the step of replacing at least a portion of said first set of solutions with niching comprises the steps of:

subdividing said first set of solutions into a plurality of sub-sets;

comparing one of said third set of solutions with members of one of said sub-sets; and

replacing one of said members of said sub-sets with said one of said third set of solutions if said one of said third solution set is superior to said members of said one of said sub-sets.

36. (Original) A method for optimizing a solution set as in claim 33 wherein each of said sub-sets have between 15-25 members.

37. (Original) A method for optimizing a solution set as in claim 33 wherein the step of creating a second set of solutions comprises using a selection operator to select members from said first solution set, said selection operator comprises one or more operators chosen from the group consisting of a tournament selection operator, a crowding operator, a pre-selection operator, a clustering operator, and a fitness sharing operator.

38. (Original) A method for optimizing a solution set for a problem comprising the sequential steps of:

a) generating a first set of solutions comprising a plurality of members;

b) creating a second set of solutions using a selection operator to select a second set of members from said first set of solutions, said selection operator using a fitness function to evaluate said members of said first set, said second set of members having a high fitness;

c) building a network to model a probability distribution of said second set of solutions, said network representing a plurality of merged variables as a single variable, said network representing dependencies and independencies between at least two of said merged variables over a plurality of hierarchical levels;

d) using said model to generate a third set of solutions;

e) replacing at least a portion of said first set of solutions with said third set of solutions to create a new solution set, said replacing comprising niching to preserve diversity among members of said new solution set; and

f) evaluating said new solution set to determine if completion criteria have been satisfied, and if said completion criteria are not satisfied replacing said first set of solutions with said new set of solutions and repeating the steps of b) through e) until said completion criteria are satisfied.

39. (Original) A computer program product for optimizing a solution set, the program product comprising computer executable instructions stored on a computer readable medium that when executed by a computer cause the computer to:

a) create a second set of solutions using a selection operator to select members from a first set of solutions;

b) build a probabilistic model fitting said second set of solutions, said probabilistic model allowing for multiple variables to be merged together to form a single new variable, and said model allowing for relationships to be modeled between at least a portion of said single new variables over a plurality of hierarchical levels;

c) using said model to generate a third set of solutions;

d) replacing at least a portion of said first set of solutions with said third set of solutions to create a new solution set; and,

e) evaluating said third set of solutions to determine if completion criteria have been satisfied.

40. (Original) A computer program product as in claim 39, wherein the program instructions when executed to cause the computer to build said probabilistic model cause the computer to build a Bayesian network that comprises local structures to model dependencies and independencies between at least a portion of said merged variables.

41. (Original) A computer program product as in claim 39 wherein the program instructions when executed further cause the computer to communicate with at least a second computer, and to cause the at least a second computer to execute one or more of the program product instructions.

42. (Original) A computer program product as in claim 41 wherein the program instructions when executed cause the computer to communicate at least a portion of said first set of solutions to the second computer for processing by the second computer.

43. (Original) A computer program product as in claim 39 wherein the program instructions when executed cause the computer to communicate with the second computer to communicate instructions to the second computer for building at least a portion of said probabilistic model.

44. (Original) A computer program product as in claim 39 wherein the program instructions when executed further cause the computer to replace at least a

portion of said first set of solutions with said third set of solutions using a niching method to preserve diversity of solutions in said new solution set.

45. (Original) A computer program product for optimizing a solution set, the program product comprising computer executable instructions stored on a computer readable medium that when executed by a computer cause the computer to:

a) create a second set of solutions using a selection operator that selects members from a first set of solutions;

b) build a probabilistic model that fits said second set of solutions;

c) use said model to generate a third set of solutions;

d) replace at least a portion of said first set of solutions with said third set of solutions to create a new solution set using niching to preserve diversity among said new solution set;

e) evaluate said third set of solutions to determine if completion criteria have been satisfied, and if said completion criteria have not been satisfied to replace said first set of solutions with said new set and to repeat steps a)-e).

46. (Original) A computer program product as in claim 45 wherein the program instructions when executed cause the computer to replace at least a portion of said first set of solutions with niching further causes the computer to:

subdivide said first set of solutions into a plurality of sub-sets;

compare one of said third set of solutions with members of one of said sub-sets; and

replace one of said members of said sub-sets with said one of said third set of solutions if said one of said third solution set is superior to said members of said one of said sub-sets.